Memory Dump Page

Local Station Application

R. Goodwin

Thu, Jul 31, 2003

Function

This application is a venerable workhorse of Local Station diagnostics, maintenance and debugging. It displays 64 bytes of memory from either the local node or from any set of up to 8 nodes, all updated at 15 Hz. This makes it possible to actually see what is happening in memory and thereby acquire a more complete understanding of the internal operation of system software. It allows setting memory bytes, words, or longwords by keyboard entry. It can transfer memory blocks of any length from any location in any node to any other location in any node. It can also fill a block of memory with any longword pattern, and it can dump a block of memory in standard S-record ascii format to any node's serial port.

Display layout and tour

```
M7MEMORY DUMP     07/31/03 1000
0572:000788 0307 3110 0047 133F
0509:000788 0307 3110 0047 1300
0614:000788 0307 3110 0047 1300
0614:000F30 C008 0000 1385 0000
0615:000788 0307 3110 0047 1300
0625:000788 0307 3110 0047 1300
0625:000790 4910 0074 4910 0076
0590:000788 0307 3110 0047 1300
ACCESS BY WORDS
RESET NODE<    >
MW 05E9:0041A000  05E9:0041A600
                       00000200
MB      00000000  0590:60001234
                       00000004
```

This snapshot shows a somewhat arbitrary example of the Memory Dump page application display. It shows 8–16 bytes of memory from the 6 different nodes, focussing on the current time-of-day found at address `0x00000788`. (Note that the time-of-day is well synchronized, as it is multicast on the network every minute by `node0616`, which has access to a network time server based upon GPS time. In this example, `node0572` shows that it does not have a Tevatron clock signal; thus, its time-of-day is based upon its own crystal rather than 15 Hz interrupts from the accelerator clock. Like all other nodes, it receives a time-of-day update via multicast every minute. The status code `7` in the second character position of the top line reflects the fact that `node0572` is running at 12.5 Hz; hence, its replies do not keep up with the expected 15 Hz replies. Under more normal conditions, that status character is blank.

Address entry

Up to 8 different addresses can be entered on the page. When an address is entered, that line and all of the following lines are automatically set to show consecutive memory data. So set different node/address combinations on different lines from the top down.

Bytes/words/longs access mode

The data bytes presented on the page are accessed by words, so that `MOVE.W` instructions are executed in the relevant contributing nodes to collect the data. Another option is access by bytes, so that `MOVE.B` instructions are used instead. A third option is access by longs (32 bits), so that `MOVE.L` instructions are used, for which the display format adjusts to show two long words per line, with 4 digits of node number and all 8 digits of address shown. Normally one wouldn't expect it would make any difference in how memory is accessed, but some hardware is designed to be accessed in limited ways. A keyboard interrupt in the first 18 columns of this line sequences through the access types `BYTES`, `WORDS`, and `LONGS`.

Memory settings

One can enter new data to be stored at the memory locations displayed. For bytes or words access, the size of the setting can be one or two bytes, or it can be one word. To enter a word, type the new value over the data value that is currently displayed at that address. (The display will not overwrite while the cursor remains in the field.) With the cursor still positioned immediately to the right of the value just typed, press the keyboard interrupt button. Expect to see the new value displayed in place of the old one. The setting of the new value word will be set as two bytes if the display uses byte access and as a word if the display uses word access.

To make a single byte memory setting, type the two digits of the desired setting in place of the current byte value and use two spaces to cover up the other byte of the word on the display. Interrupt just after the word and only the single byte value will be set.

When memory is being access by long words, and two long words are shown per line, one can make long word settings by interrupting with the cursor immediately after the 8-digit value.

Another way to make settings of memory data uses the features described next.

Copy, fill, and dump

The next four lines allow for copying memory blocks, filling them with constant data, or dumping them to a serial port. The first one is set to copy `0x200` bytes of memory words (“`MW`”) from address `0x0041A000` in `node05E9` to `0x0041A600` in the same node. The second one is set to copy 4 bytes of `0x00` to address `0x60001234` in `node0590`. The difference in specifying the fill option rather than the copy option is denoted by the absence of the colon that must be present in the first `nnnn:address` field. A keyboard interrupt action with the cursor in the area of either pair of lines will cause the indicated operation to be performed. The long word option for access (“`ML`”) is also available.

Other options available on these lines are `DB`, `DW`, and `DL`, which cause bytes/words/longs of data from a memory block to be encoded into standard Motorola S-record ascii format and spooled to the serial port of a selected node. To select the dump option, interrupt with the cursor in the first column, under the “`M`”, and it will be changed to “`D`”. Another interrupt switches back to the move/fill option. Byte, word, or long access for these operations is similarly selected by interrupt in the second column. The dump option provides a way of archiving memory data on the disk of a pc or host computer without using the network.

Reset node

One can reset a target node by entering its node number in the `RESET NODE` field and interrupting. This field is always preset to blanks upon entry to the page as a caution. To be effective, the system needs to be running well enough to accept the Classic protocol setting. This feature is most often used to cause a node to boot up the latest version of system code.

Pointers

When examining software data structures, it is sometimes desirable to follow a pointer chain, or linked list. A facility for doing this is included on the Memory Dump page. An interrupt with the cursor located at either of the two middle hex digits of the upper word of a memory address that is currently displayed (as two words in the data area when in access mode `BYTES` or `WORDS`, or as a long word in access mode `LONGS`) causes the contiguous region displayed (usually 64 bytes) that includes that address to be replaced by the same size region starting at the address indicated. *This is much easier to do than say.* Additional interrupts under addresses displayed can follow a linked chain of data structures. Each time this happens, the address (and node) is saved in an internal table. One can navigate through this table by interrupts with the cursor toward the right end of the line immediately following the 8 data lines, where an address will be shown. Interrupt under the left half of the address to back up in the table; interrupt in the right half to move forward through the pervious saved addresses.

If one backs up all the way to the start, the displayed address vanishes. Up to 100 addresses are saved internally, all of which are lost when leaving the page.

Freeze display

When the display is being updated at 15 Hz, it is sometimes difficult to catch values that change often, or to correlate them with other changing values. To assist in viewing a snapshot of the memory displayed, press the Hex button on the Local Console. The contents of the screen will be frozen as of that moment, and the "`NO UPDATE`" message will be displayed. Release the Hex button to resume normal updating. Note that while the display is frozen, the data is still being collected; it is simply not being updated on the display. Another way to make snapshots is via the serial port printing option below. (This switch-based, not LED-based, feature does not currently work when using "Page G" emulation.)

Volts units

If the data words being displayed are `A/D` readings, they can be shown in voltage units by selecting the Volts button on the Local Console. This assumes that the scaling used is ±10 volts full scale. The value `0x8000` corresponds to –10 volts, and the value `0x7FFF` represents +10 volts (actually +9.9997 volts). Zero volts is `0x0000`. Settings do not work in Volts units.

Floating point units

Selecting the Eng button on the Local Console displays each 4 bytes of memory data as a floating point value, assuming it can recognize them as a valid IEEE single precision floating point value in a reasonable range. If it is not reasonable, then it defaults to displaying a long word in hex. Settings do not currently work in this mode, although it could be added.

Entering addresses

To enter addresses in the memory data part of the display, several conventions are in use. Typing an address alone at the start of a line, overwriting the node number, causes only the address part to be changed. But take care to terminate the address with a space, so that additional digits starting at the current cursor position are not accepted as part of the new address. If one merely changes the node digits, leaving the address alone, then only the node will be changed to match what has been modified. If both node and address are entered, both parts will be accepted.

The small local displays have only 32 characters per line, so there is not enough room to show a complete node#, a 32-bit address, and 4 data words with enough spacing to make it readable. Since it is common to use addresses within the 24-bit range `0x000000`—`0xFFFFFF`, or to use node#s in which the hi byte is always the same, there is a toggle between two options of displaying the `node:address` fields. To alternate between the two options, interrupt in the first column of the second row—under the first character of the first node#.

If the interrupt is given with the cursor in the first column of any data line after the first, then the address accepted is taken from the previous line but with the node number incremented by one. If the cursor is on the last data line, the current address is not changed; only the node number is incremented. And if the cursor is not on the last line, then the cursor is moved to the start of the next line. This makes it easy to look at the same 8 bytes of memory in successively-numbered nodes.

The raise and lower push buttons can be used to move forward and backward through memory. When the raise button is pressed, the address on the current line is advanced by the number of bytes of data that is displayed on that line plus the following lines, and the addresses are forced to be contiguous memory in the same node as on the cursor line. This allows one to scan through a region of memory displaying up to 64 bytes at a time. If the cursor is on a line *other* than one of the eight data lines, memory addresses are advanced 64 bytes at a time just the same as if the cursor were on the first data line.

Smart Rack Monitor (SRM) access

Local stations connect to SRMs via Arcnet. A special range of node numbers in the range 7AA0–7ABF is reserved for this purpose. Arcnet nodes use node#s, allowing for a maximum of 32 SRMs on a single Arcnet network. Linac control stations have 1–6 SRMs connected via Arcnet. Although the memory request protocol is different for SRMs than for local stations, special support is built into the memory dump page to permit viewing Arcnet-connected SRM memory in a natural way.

Small Memory Dump

This is a system function and is therefore available independent of what application page is running. Still, it affords one more line of 8 bytes of memory data that is available for display using the bottom line of the small screen display. (The easiest way to find it is to press the Home key and then the Up arrow key.) Enter an address at the start of the line; the node number is not required, as this line only displays local memory in bytes. If a bus error is found, the data bytes are displayed in inverse video. Local memory settings can be made in the same way as is done via the Memory Dump page. The raise and lower push buttons can be used to move through local memory 8 bytes at a time while the cursor is on the bottom line.

Printing to the serial port

This is also a system function, but it bears including here. Any application page display can be written to the local serial port by a keyboard interrupt under the page title on the top line (actually the 3rd through the 18th column). Sometimes it's useful to save copies of the memory display for later perusal.

Error codes

In response to data requests, an error code may be returned. The Memory Dump page displays this code on the top line in the second column (between the current page number and the page title) if it nonzero. Only a few codes are commonly seen. An 8 code means that some node whose data is requested in the list of addresses on the display is not returning any data in response to the request. This may mean that it's down or in some other way inaccessible via the network. A 7 code means the replies are tardy, but that data was received from that node at least once since the memory data request was last issued, so the node may not be down. (During the time that no data is received from an addressed node, the underlying system will re-send the request to the node every 2 seconds in case it comes up.) This code appears often intermittently when nodes are not running in synchronization. The other code that may appear is a 4 code, which means a target node experienced a bus error when accessing the memory data. It is described next.

Bus errors

The system has no means of predicting whether a given address from which data is requested will produce a bus error when it is accessed. When the access is made by the system code in processing the data request, it treads carefully. If a bus error is encountered while accessing the memory requested (by bytes, words, or longs as selected) an error 4 is returned with the data in response to the request.

Internal matters

Memory copy block sizes

When moving memory blocks around, it is done with 1024-byte packets at 15 Hz. This means that 100K bytes of memory—which happens to be about the size of the system code—can be transferred in about 7 seconds. In the case of memory transfers to Arcnet-based SRMs, the maximum packet size is 480 bytes, in conformance with the smaller maximum frame size available with Arcnet.

SRM task names

Normal SRM communications use a simple Acnet header-based protocol that is addressed to the destination task name SRMD. But to download a new version of the SRM system code, the task name NEWV must be used. The memory dump page program checks for the special case of a new version download by the range of memory specified in the memory copy. If it is in the range used for the SRM system code, then it uses the NEWV task name; otherwise, it uses SRMD. After the transfer is complete, in the case that NEWV was used, a reset message is sent using the task name REST. This resets the SRM and causes it to begin execution of the new version that was just downloaded.

Access to data from other nodes

Every data request or setting request includes a node number along with the memory address being referenced. The application program pays very little attention to whether the node number used is the local node or not. The system code that is invoked to make a data request does whatever is required to accomplish the task. If network communication is required, then it uses the network; otherwise, it does not. The job of writing application programs would likely be much more difficult if this were not the case. Again, to the user, this makes no difference at all. She needs only to specify the nodes of interest.

The PAGEMDMP executable program file size in the IRM 68040-based nodes is about 12K bytes. The object file size in PowerPC-based nodes is about 19K bytes.